

SOFTWARE HELP

Using PC as a guitar amp

Q I play the guitar, but my amp isn't very popular with the neighbours and it doesn't have a headphone socket. Is there any way I can use my PC as a guitar amp for late-night practice?

Craig Phelps

A There's a wide range of virtual guitar amp software available, with prices ranging from free to hundreds of pounds. Some run as standalone applications, while others are VST plug-ins that run inside VST-compatible recording software. Standalone versions are easier for rehearsal, but VST plug-ins can be used for both rehearsal and recording.

In both cases you'll need a sound card that's able to deliver low-latency audio performance. Latency is the time it takes for a sound to travel into your PC, through the software you're using and out to your speakers or headphones. Low latency is vital so you can hear your playing without any delay. This is possible using Windows' standard WDM drivers, but our experience is that ASIO drivers are more reliable. ASIO drivers are standard on sound cards designed for musicians, but they're rare on general-purpose sound cards and integrated motherboard audio.

Most sound cards have microphone and line inputs, but a guitar's output level is somewhere between the two. It is possible to use a line input and crank the volume right up, but the tone will be thin. It's much better to use a sound card with a dedicated guitar input (sometimes called an instrument input). Tapco's Link.USB (*What's New, Shopper* 230) costs £69 from www.dv247.com and includes two guitar inputs, ASIO drivers and excellent VST-compatible Mackie Traktion recording software.

The most affordable guitar amp simulator software is AmpliTube Uno, available free from www.audiomidi.com/files/amplugz. It runs as a standalone application using either ASIO or WDM drivers. Its controls are straightforward and quality is excellent, with processing taken from the full version of AmpliTube, which costs



◀ Native Instruments' Guitar Rig 3 is an expensive but luxurious way to turn a PC into a virtual guitar amp

around £180. We had difficulties getting it working reliably with our setup, but seeing as it's free, there's no harm in trying it out.

AmpliTube Live is the mid-price program from the same range and costs £39 from www.turnkey.co.uk. It includes a wider range of modelled amps and effects, and works in standalone and VST modes. For the less budget-conscious, Native Instruments' Guitar Rig 3 includes authentic re-creations of 21 classic guitar amps, tons of effects and a hardware foot controller that doubles as a USB sound card. It retails for £349, but version 2 is currently available for £200, with a free upgrade to version 3, from www.dv247.com. It works as a standalone application or as a plug-in, although we haven't had a chance to test its performance.

Whichever software and hardware you choose, you'll need to spend some time experimenting with latency settings, located alongside where you select which sound card the software addresses. The lower the setting, the less delay there will be between striking a string and hearing a note, but go too far and clicks or glitches will start to appear. Settings from 3 to 6ms (sometimes quoted as 128 to 256 samples) are indistinguishable to zero for most people, but you might find that a higher value is necessary for reliable performance.

If you're using a VST plug-in, you'll need to add it as an insert effect to a mixer channel in your host recording software, and then activate the audio input for that channel. **BP**

YOUR EXPERTS

Contact Software Help for software advice and answers

Kay Ewbank

Database developer and productivity expert Kay Ewbank offers help with office applications.



Ben Pitt

Professional musician and keen photographer Ben Pitt gives advice on using creative software.



softwarehelp@computershopper.co.uk



◀ AmpliTube Uno is a free guitar amp simulator with simple controls and great sound quality, but setup can be tricky

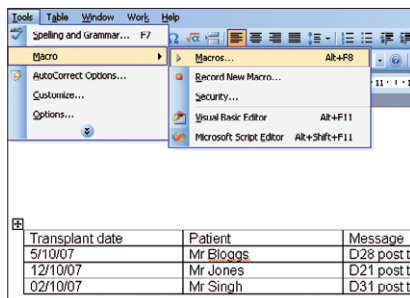
I need a Word table for my patients

Q I am a doctor and have a list of patients arranged in a table in Word. In various cells (among other text) I have comments such as "D4 post transplant", indicating that four days have elapsed since the patient's transplant. I currently change this manually each morning to reflect an additional day.

I was wondering if it were possible to automate this, in effect subtracting two dates (Today – previous date) from each other and returning the answer in an integer. I am aware that this is very easy to do in Excel, but I want to stay within Word if possible. Ideally it would best be done in a field, but I don't know how feasible this is.

Kinesh Patel

A There are a number of ways of doing this. You could use a field code, with the most likely choices being either a database field or a docvariable field. The disadvantage of using a database field is that it would involve using the Jet database engine, and you want to stay completely within Word. The disadvantage of the docvariable field is



▲ Use the macro on the right to update entries in a table automatically

that it is hidden from the 'normal' Word interface and changes need to be made using a macro. For this reason, a method where a macro can be used to update the text in a table is preferable, because the document is then equally usable without the macro.

I do think using Excel might be even better, though. You can format an Excel document to look just as good as a Word document, and it's a lot better at arithmetic.

However, if you're still keen to use Word, the macro below assumes that the patient details are stored in a table something like this:

TRANSPLANT DATE	PATIENT	MESSAGE
5/10/07	Mr Bloggs	D28 post transplant
12/10/07	Mrs Jones	D21 post transplant

The macro is fairly simple:

```
Sub daysafter()
Dim strFirstDate As Date
Dim elapsed As Integer
Dim mytable As Table

Set mytable = ActiveDocument.
Tables(1)
Set aColumn = mytable.Columns(1)
For Each acellInaColumn.Cells
If acell.RowIndex = 1 Then
'title row
Else
strFirstDate = CDate(CleanString
(acell.Range.Text))
thisrow = acell.RowIndex
elapsed = DateDiff("d",
strFirstDate, Date)
mytable.Cell(thisrow, 3).Range.
Text = "D" & elapsed & " post
transplant"
```

```
End If
Next acell
End Sub
```

This assumes that the table is the first one in the document – if it isn't, change the digit 1 to the correct value in the line reading

```
Set mytable = ActiveDocument.
Tables(1)
```

The macro assumes you will enter the transplant date manually. The body of the macro works through column 1 of the table a cell at a time:

```
For Each acellInaColumn.Cells...Next
acell
```

The first row of the table is skipped because it holds the titles for the columns. From then onwards, the value in column one is converted into a date using:

```
strFirstDate = CDate(CleanString
(acell.Range.Text))
```

Range.Text returns the text in the cell, but it gets pulled in with a paragraph mark, so CleanString is used to remove all non-printable characters, including the paragraph marks. The text is then converted to a date using CDate. DateDiff is then used to find the number of days between the date in column one and the current date (given by the function Date). You can make DateDiff find a variety of date differences depending on what date interval you specify – 'q' for quarter, 'w' for weekdays, 'ww' for weeks and so on. I've used 'd' for days. The resulting value is then placed into column 3 of the table along with the text 'D' and 'post transplant'. **KE**

How do I set up a Word index using XE fields?

Q I'm using Word and wondered whether the 'invisible' XE entries might be useful for creating an index for export to Access? I'd also like to know what combination of keys produces the brackets and the 'XE'. I could then work on data on a non-Windows machine.

Graham Mangham

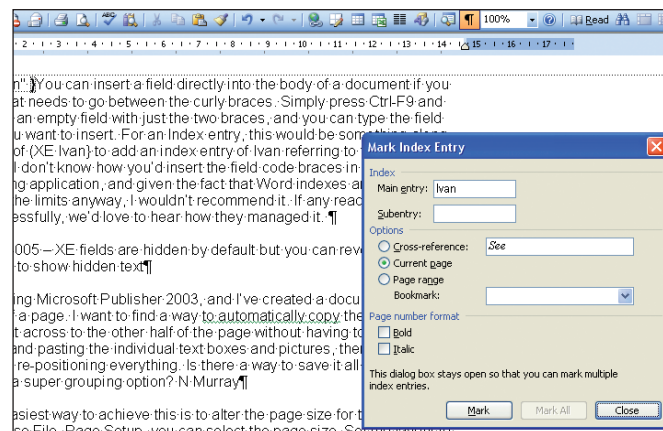
A The XE field is used to set up an index. We think Microsoft Word is a good word processor, but using it for very long documents and, in particular, relying on it for holding complex multi-chapter documents is often a recipe for heartache and problems. For this reason, creating an index in Word is something that we'd be loath to do. That caveat in place, you can use an XE field to set up an index.

You can insert an index field in several ways. The easiest way is from the Insert menu. Choose the Field option, then choose Index from the options on offer. As you mention, the XE fields are formatted as hidden text. If you want to see the field, click Show/Hide (the

button that looks like a paragraph marker or back-to-front 'P' on the formatting toolbar).

You can insert a field directly into the body of a document if you know what needs to go between the curly brackets. Simply press Ctrl-F9 and you'll get an empty field with just the two brackets, and you can type the field codes you want to insert. For an Index entry, this would be

something along the lines of {XE Shopper} to add an index entry of Shopper referring to the current location. We don't know how you'd insert the field code brackets in another word processing application and, since Word indexes are likely to be pushing the limits, we wouldn't recommend it. If any readers have done this successfully, we'd love to hear how they managed it. **KE**



◀ XE fields are hidden by default but you can reveal them by choosing to show hidden text

Creating DVD presentations

Q I want to produce PowerPoint-style presentations incorporating full-screen video, but my laptop is too old and slow for high-quality video. Is it possible to make an interactive presentation to play on a standard DVD player?

Audrey Harrison

A Most low-cost DVD-authoring software would struggle with this task, but Sony DVD Architect Studio (What's New, Shopper 230), available for £30 including VAT from www.amazon.co.uk, takes it in its stride. A trial version is available from www.sonycreativesoftware.com/download.

DVD Architect Studio lets you combine text and graphics freely on a menu page; simply right-click and select Insert Text or Insert Graphic. Comprehensive text formatting and alignment tools are included, but be careful to stay inside the dotted lines, as most TVs and some projectors crop the edges of a DVD player's image. To change the background, double-click a background from the tab at the bottom or right-click the preview, select Set Background Media and browse to your file.

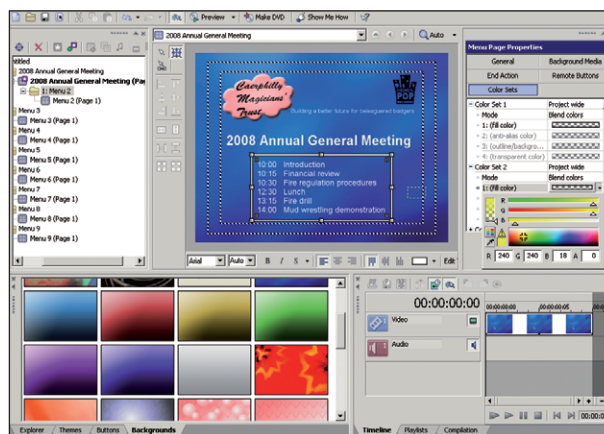
You create a second slide by right-clicking the Project Overview panel on the left and selecting Insert Menu. To link the two slides together, drop the icon for the second one on to the first. This will create a DVD button on the first slide, linking to the second slide and consisting of a thumbnail image and text.

However, while we want the capability of this button, we don't actually want to see it. To hide it, select its text component and delete it. Select the thumbnail, shrink it using the handles and tuck it out of the way in a corner of the slide. This thumbnail is blank by default, but it will still have a visible highlight to show that it is selected. To remove this, select Color Sets from the Properties panel on the right, and for Color Set 1, select Project wide from the drop-down list and change the Fill colour to fully transparent. Under Mode, select Blend colors. Do the same for Color Set 2.

To add full-screen video to your presentation, import a video file and use it just as you would in any normal DVD project by dropping it on to a menu. This creates a button with a thumbnail of

the video. To change it into a blank button, select it, click the Media tab in the Properties panel and, under Thumbnail Media, select Remove. Double-click the button to open the video and select the End Action tab in the Properties panel. Under Destination, select the menu you want to display after the video has ended. Alternatively, if you want to be able to interrupt the video to jump to the next slide, use the video clip as the background media for a menu. However, you won't be able to pause its playback.

You can test your presentation at any time by clicking the Preview button at the top and using the virtual DVD remote control. You'll be able to progress to the next slide or video simply by clicking the Enter button. **BP**



◀ With a bit of lateral thinking, Sony DVD Architect Studio is highly effective at creating presentations on DVD-Video disc

HTML report problem in Excel

Q I have a number of HTML reports that I can open in Excel. However, when I open each one I get a separate workbook. I would like to open them all in the same workbook on separate sheets. Is this possible? The reports are generated from Oracle database 10G and are AWR reports.

Alan McKeown

A There isn't a one-stage way of doing this, because of the way Excel handles HTML reports. The best way is to open the reports separately. Select one of the reports and, from the Edit menu, choose the option Move or Copy Sheet. You'll see a dialog box where the first choice is the book (workbook) to which you want to move or copy the sheet. The default choice shown is the

name of the current workbook, but if you click on the drop-down box you'll see the names of the other open workbooks. Select the book to which you want to add the report you're currently viewing. You then get the choice of where to put the new sheet in relation to the existing sheets.

You can put the current sheet in front of the existing sheets, in which case you select the name of the sheet you want the current one to be placed before. Alternatively, choose '(move to end)' to place the current HTML report at the end of the sheets. You can then repeat this process for each report you open.

If you don't want to do this manually, the macro below opens all the HTML documents in the folder C:\myHTML, and puts them all into a single workbook:

```
Sub html()
    myname = Dir("c:\myhtml\*.htm")
    Do While myname <> ""
        Workbooks.Open (myname)
        myname = Dir
    Loop
    countbooks = Workbooks.Count
    For i = countbooks To 2 Step -1
        Workbooks(i).Activate
        mysheet = Workbooks(i).Sheets(1)
        Name
        MsgBox (mysheet)
        Sheets(mysheet).Move After:=
        Workbooks(1).Sheets(1)
```

Next
End Sub

The macro first finds and opens all the documents in the directory C:\myhtml that have a *.htm extension. This is done in the code:

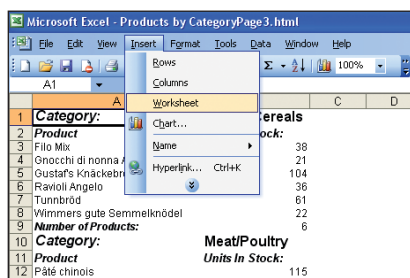
```
myname = Dir("c:\myhtml\*.htm")
Do While myname <> ""
    Workbooks.Open (myname)
    myname = Dir
Loop
```

The line:

```
myname = Dir
```

gets the next item in the directory.

Next, countbooks is set to hold the number of open workbooks. The next loop then activates each workbook in turn, starting with the final workbook. The reason for starting at the end rather than the beginning is that when you move a workbook into the first workbook, it disappears from the open workbooks list, and Excel eventually gets into a pickle because you're asking for the equivalent of workbook four, for example, when there is no longer a workbook four. If, however, you start with workbook four, and move it into workbook one, then the next workbook you ask for is workbook three, which is still there. **KE**



▲ You can add sheets even on workbooks containing HTML pages

Sage advice required

Q I use Microsoft Access 2003 to link to data in Sage Line 50 12 using the Sage ODBC driver. The tables in Sage to which I link are Purchase_Orders and POP_Item, which contains all the individual items ordered on each Purchase Order. I have a problem that occurs when line items are deleted from purchase orders. These deleted line items don't appear in Sage but they still appear in the linked table POP_Item in Access, which means any reports I derive from this data are incorrect.

Is this a limitation in the Sage ODBC driver, and is there another way to get around this problem? It is possible to compress the data for Purchase Orders in Sage, which seems to remove these deleted items, but this isn't a satisfactory solution, especially as I am not the main user of this module in Sage and not always aware when items have been deleted.

Leslie Robinson

A There are two ways that Sage marks records to show that they have been deleted. If the main entire purchase order record has been deleted, the order_number field in POP_Item is changed to -1. The POP_Item table uses a line number field called Item_Number. It seems that if an order line is deleted from an order, the lines above that item are renumbered downwards, so if item number three is deleted, then what used to be item four is renumbered as item three, item five gets renumbered as item four and so on. This means that you can find which items are deleted by looking for pairs of lines where the order number and item number are the same.

There are three possible ways to deal with this, all of which involve local copies of the tables rather than using the linked tables, as the technique changes the data in the tables.

The first method is to delete the records from a local copy of the tables. To do this, you first need to identify the records that Sage has marked as deleted. Records where the order has been fully deleted are easy to deal with; if the order number is -1, you delete that record from the POP_Item table.

It's harder to deal with the items where a line item has been deleted. You need to find the records where the order number and item number are the same, then delete the record that occurs earlier in the record sequence. You

can identify which record came earlier because the POP_Item table has a field called This_Record that contains sequential numbers showing the order in which records were created.

The easiest way to look for the first record in these matching pairs is to make the This_Record field the primary key so that you know the table is ordered in record creation order. Then you start at the end of the table and look backwards; that way, the second match in any pair you find will be the earlier number, and the one deleted by Sage. If you start at the beginning of the table and look forward you'll need to delete the record before the one you're currently on whenever you find a match.

```
Sub sage()
    Dim dbsSageDemo As Database
    Dim rstPopItem As Recordset

    Set dbsSageDemo = OpenDatabase(
        "SageDemo.mdb")
    Set rstPopItem = dbsSageDemo.
        OpenRecordset("Pop_Item",
            dbOpenDynaset)

    With rstPopItem
        .MoveLast
        .MoveFirst
        Do While Not .EOF
            If !order_number = -1 Then
                .Delete
                .Requery
            Else
                .MoveNext
            End If
        Loop

        .MoveLast
        thisord = 0
        thisitem = -1
        Do While Not .BOF
            If !item_number = thisitem Then
                If thisord = !order_number Then
                    .Delete
                Else
                    thisord = !order_number
                End If
            Else
                thisitem = !item_number
                thisord = !order_number
            End If
            .MovePrevious
        Loop
```

```
End With
rstPopItem.Close
dbsSageDemo.Close
End Sub
```

The second option is to change the codes in the records where you meet a pair so that the item number is changed to be -1. You can then create a query to see all the records that have been identified as deleted. The code for doing the changes is quite similar to the code for deleting:

```
Sub sage2()
    Dim dbsSageDemo As Database
    Dim rstPopItem As Recordset

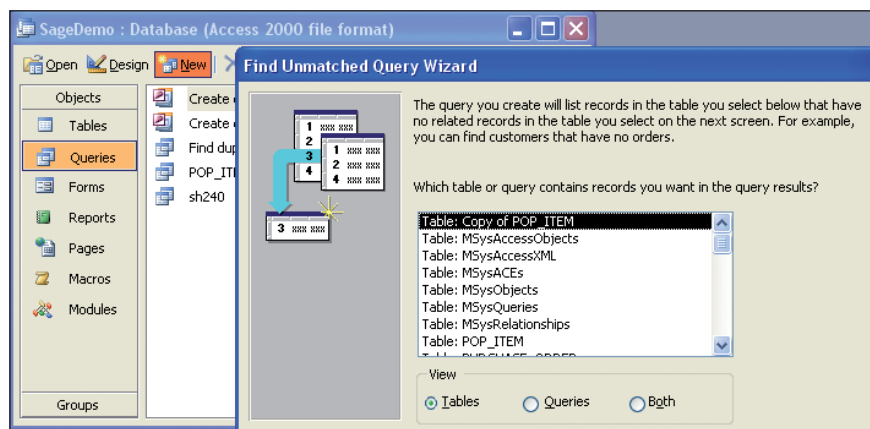
    Set dbsSageDemo = OpenDatabase(
        "SageDemo.mdb")
    Set rstPopItem = dbsSageDemo.
        OpenRecordset("Pop_Item",
            dbOpenDynaset)

    With rstPopItem
        .MoveLast
        thisord = 0
        thisitem = -1
        Do While Not .BOF
            If !item_number = thisitem Then
                If thisord = !order_number Then
                    .Edit
                    !item_number = -1
                    .Update
                Else
                    thisord = !order_number
                End If
            Else
                thisitem = !item_number
                thisord = !order_number
            End If
            .MovePrevious
        Loop

    End With
    rstPopItem.Close
    dbsSageDemo.Close
End Sub
```

You'll then need a query pulling records from the Pop_Item table, where the selection criteria is that either the order or the item number is -1. Once you're happy that the correct records have been selected, you can delete them manually.

The final method would be to modify this to avoid deleting the records. You'd use the code to set the item number on matching pairs to be -1, and create your selection query to identify such records. You could then use a Find Unmatched query to find those records that hadn't been identified by the previous query. So you'd use the Find Unmatched Query wizard, select Pop_Item as the table containing the records you want in the query results, then specify the query showing the 'deleted' records as the source of the related records. The matching field would be This_Record, and your results would then show only those records from Pop_Item that hadn't been pulled out in the query identifying records with -1 in either the order or item number. KE



▲ Use the Find Unmatched Query wizard to identify records that haven't been marked for deletion